UCRL-CONF-203679

Progressive Compression of Volumetric Subdivision Meshes

Daniel Laney, Valerio Pascucci

April 21, 2004

3D Data Processing, Visualization, and Transmission
Thessaloniki, Greece
September 7, 2004 through September 9, 2004

Progressive Compression of Volumetric Subdivision Meshes

Daniel Laney Valerio Pascucci
Lawrence Livermore National Laboratory
dlaney@llnl.gov pascucci1@llnl.gov

Abstract

We present a progressive compression technique for volumetric subdivision meshes based on the slow growing refinement algorithm. The system is comprised of a wavelet transform followed by a progressive encoding of the resulting wavelet coefficients.

We compare the efficiency of two wavelet transforms. The first transform is based on the smoothing rules used in the slow growing subdivision technique. The second transform is a generalization of lifted linear B-spline wavelets to the same multi-tier refinement structure. Direct coupling with a hierarchical coder produces progressive bit streams. Rate distortion metrics are evaluated for both wavelet transforms.

We tested the practical performance of the scheme on synthetic data as well as data from laser indirect-drive fusion simulations with multiple fields per vertex. Both wavelet transforms result in high quality trade off curves and produce qualitatively good coarse representations.

1 Introduction

The processing and representation of data on regular grids has advanced substantially in recent years. Large scale scientific simulation codes that generate data on regular grids have been coupled with advanced wavelet compression systems and cache coherent streaming systems. However, a substantial amount of data is generated using finite element or finite volume techniques that have meshes with enhanced or reduced connectivity and with vertex positions that change with time.

This paper presents a progressive compression framework for volumetric data created by scientific simulations that is a first step to a full progressive system for arbitrary finite element grids. Such a system involves three main tasks:

1. **Mesh Remapping:** an efficient and robust method of taking arbitrary finite element meshes with associated fields and producing a conforming mesh with subdivision connectivity.
2. **Compression:** compression of the data on the subdivision mesh with controllable error tolerances.
3. **Flexible Visualization**: a visualization system that supports view-dependent rendering to enable interactive navigation of large data sets.

This paper presents a compression system for slow growing subdivision volumes with associated field values. The slow growing refinement rules coupled with progressive wavelet transforms enable progressive reconstruction of scientific data, and can be used to produce view dependent approximations. The remapping algorithm used here is quite simple, and not a general solution. However, remapping algorithms are still under active research, and currently no general solution is known to the authors.

2 Previous Work

Wavelet transforms have been used to construct multiresolution representations of scalar volume data for rendering and compression [8, 9, 10, 3, 4]. Wavelets have been used to produce progressive representations [16] and time-varying volumes [18] for volume data.

Subdivision surfaces have also been combined with wavelet techniques to produce compressed progressive representations. Khodakovsky [5] presented a complete system for triangle meshes. The system used the MAPS algorithm [6] to construct a mesh with subdivision connectivity from unstructured mesh data. Bertram [1] generalized the lifting scheme of Sweldons [15] to Catmull-Clark [2] surface subdivision and applied it to isosurfaces extracted from large scale simulation data. Valette *et. al.* [17] have proposed a subdivision wavelet scheme for irregular meshes that does not require a base mesh with subdivision connectivity. The present work requires such a base mesh and would be difficult to extend in the same way.

Linsen [7] used a similar approach to the present paper, generalizing B-spline wavelets to $\sqrt[n]{2}$ subdivision for regular grids. The wavelets were factorizations of the lifted wavelets of Bertram [1] to the multi-tier refinement. The coefficients were thresholded and small numbers of coefficients were used to produce approximations to scientific data sets defined on regular grids.

3 Slow Growing Subdivision

The slow growing subdivision scheme [11] generalizes to arbitrary dimension, but only the three dimensional case is treated in this paper. This section provides a brief overview of the technique and relevant notation.

Regarding only the connectivity of a mesh, the slow growing refinement rules for volumes are a factorization of the Catmull-Clark refinement rules [2] into three distinct steps. These steps are referred to as *tiers*, and labeled from 0 to 3. The subdivision is indexed by the level ℓ and tier with $(\ell, 3)$ equivalent to $(\ell+1, 0)$. In the remainder of this section, mesh elements will be described by the tier at which they exist. Thus, "tier 0" faces, means the faces of the mesh the exist at tier 0. As will be seen below, the connectivity of the mesh changes from tier to tier, making such distinctions necessary.

3.1 Refinement Rules for Connectivity

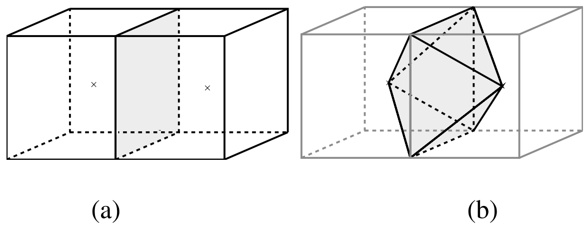

Figure 1. Cell refinement from tier 0 to tier 1. (a) Two cells in tier 0 with common face shaded gray. Their centers are marked with two crosses. (b) The new cell of tier 1 (in gray) is the union of the pyramids defined by connecting the tier 0 cell centers to the vertices of the tier 0 cells. The common face is removed from the mesh.

Figure 1 shows the construction of tier 1 cells from tier 0 cells. A vertex is inserted within each tier 0 cell. This vertex is connected to each vertex of the cell, forming a set of pyramids. Two such pyramids are shown in figure 1b. Pairs of pyramids sharing faces of tier 0 cells are merged to form tier 1 cells. The shared tier 0 faces are then removed from the mesh. The faces of the mesh can be tagged as *features*, which causes them to remain in the mesh. It also causes any subdivision rules applied to the face to utilize only the vertices of the face (i.e. effectively introducing two dimensional surface subdivision). In summary, the first step of slow growing subdivision inserts vertices in *cells* and merges along *faces* of the tier 0 mesh.

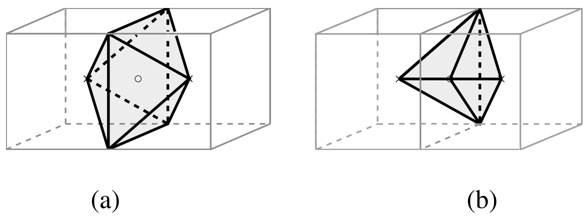

Figure 2. Vertex insertion and subdivision of tier 1 cells. (a) A vertex (indicated by a $\circ$) is inserted in a tier 1 cell. (b) The vertex is connected to each original vertex of the tier 1 cell, creating a set of pyramids. Two pyramids sharing an edge of the tier 0 mesh are shown.

The second step of slow growing refinement, from tier 1 cells to tier 2 cells begins by introducing vertices inside tier 1 cells as shown in figure 2a. Each vertex is connected to the vertices of the tier 1 cell to create a new set of pyramids. Note that the newly inserted vertex is associated with a face of a tier 0 cell. It is not constrained to lie on this face unless that face was marked as a *feature*.

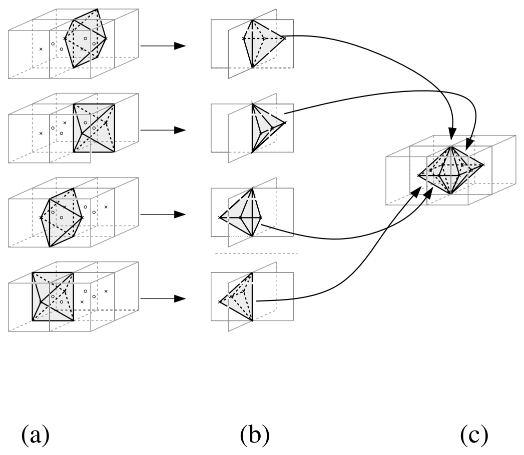

Figure 3. Creating tier 2 cells at edges of tier 0. (a) Tier 1 cells sharing a tier 0 edge. (b) Subdivision of tier 1 cells into pairs of pyramids sharing the tier 0 edge. (c) Creation of a tier 2 cell by merging the pyramids about the shared tier 0 edge.

Figure 3 depicts the merging operation that generates tier 2 cells. All newly created pyramids that share an edge of the tier 0 mesh are merged along that common edge, and the common edge is removed from the mesh. Edges can be tagged as being *features*, causing them to stay in the mesh and prohibit the adjacent pyramids from being merged. Edge features constrain subdivision to one dimension. In summary, the second step of slow growing subdivision inserts vertices corresponding to *faces* of the tier 0 mesh, and merges along *edges* of the tier 0 mesh.

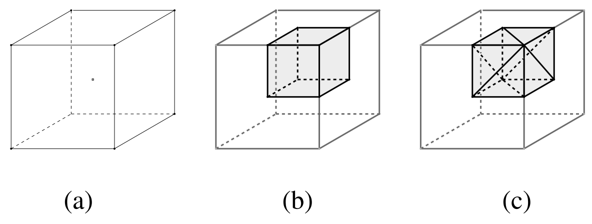

Figure 4. Creation of a tier 3 cell (equivalent to tier 0 at level $\ell+1$). (a) Tier 0 cell. (b) Tier 3 cell shown in gray. (c) Tier 3 cell showing all edges created in the subdivision process from tier 0 to tier 3, which are removed during the merge operations of tiers $1..3$.

Figure 4 shows the final step of the slow growing refinement. Vertices are inserted into tier 2 cells (corresponding to edges of the tier 0 mesh), and pyramids constructed by connecting these vertices to the vertices of the tier 2 cells. All cells sharing a vertex of the tier 0 mesh are merged, and all edges present in the tier 1 and tier 2 cells are deleted. After the last step of subdivision the connectivity of the mesh corresponds to the connectivity that would be obtained by generalizing Catmull-Clark subdivision to the tier 0 cells.

3.2 Vertex Positioning and Smoothing

All vertices are vector valued, and may contain additional components representing physical quantities associated with the vertices (e.g. pressure or temperature). A tier k vertex will be denoted by $\mathbf{v^{(k)}}$. Vertices inserted into cell cell centers are denoted by $\mathbf{v_C^{(k)}}$. Given a cell C of tier k with n vertices $\mathbf{v_0^{(k)}}..\mathbf{v_{n-1}^{(k)}}$, a tier $k+1$ vertex $\mathbf{v_C^{(k+1)}}$ is inserted at the centroid of C:

$$\mathbf{v_C^{(k+1)}} = \frac{1}{n}\sum_{i=0}^{n-1}\mathbf{v_i^{(k)}} \tag{1}$$

After connectivity is updated according to the description in section 3.1 the tier k vertices of the original set of cells are smoothed according to

$$\mathbf{v^{(k+1)}} = \alpha\mathbf{v^{(k)}} + \frac{1-\alpha}{m}\sum_{\forall \mathbf{v}\in\mathcal{E}(\mathbf{v^{(k+1)}})}\mathbf{v} \tag{2}$$

where $\mathcal{E}(\mathbf{v^{(k+1)}})$ is the set of vertices edge-connected to $\mathbf{v^{(k+1)}}$ in the tier $k+1$ mesh and m is the total number of such vertices. $\mathcal{E}(\mathbf{v^{(k+1)}})$ includes both tier k vertices and new tier $k+1$ vertices inserted according to equation (1).

4 Wavelet Transforms

A theoretical framework for constructing stable wavelets on the multi-tier refinement structure of the slow growing subdivision is not currently available. In contrast to surface subdivision methods, many configurations can occur at extraordinary vertices in volume meshes which make a spectral analysis of scaling functions difficult. Thus, in the next sections two wavelet transforms are presented and empirical results are presented with respect to their stability. The two methods are inspired by the lifting technique of Sweldons [15].

Wavelet decomposition transforms a fine scale subdivision volume to a coarse scale base mesh with all finer level vertices replaced by wavelet coefficients. Wavelet reconstruction reverses this process, reconstructing the fine scale subdivision volume.

For all tiers the decomposition is accomplished by replacing some tier $k+1$ vertices with wavelet coefficients and removing the tier $k+1$ cells incident on those vertices, producing tier 0 cells. In the following sections, the vertices removed by decomposition are denoted by $\mathbf{v_C^{(k+1)}}$ since they are tier $k+1$ vertices, but are associated with the centers of tier k cells. These vertices are replaced with wavelet coefficients $\delta^{(\mathbf{k})}$. Thus, reconstruction of tier $k+1$ is accomplished using tier k vertices $\mathbf{v^{(k)}}$ and tier k wavelet coefficients $\delta^{(\mathbf{k})}$.

4.1 Slow Growing Subdivision (SGS) Wavelets

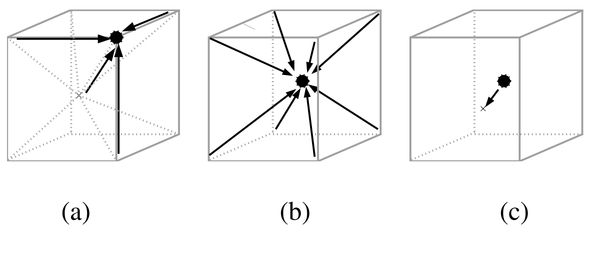

Figure 5. SGS wavelet decomposition for a set of six tier 1 cells with vertices at the corners labeled $\mathbf{v_0}^{(1)}..\mathbf{v_7}^{(1)}$ and a center vertex $\mathbf{v_C}^{(1)}$. (a) Smoothing of $\mathbf{v_i}^{(1)}$ vertices by edge neighbors. (b) Computation of centroid of vertices $\mathbf{v_i}^{(1)}$. (c) The tier 0 cell, with wavelet $\delta^{(0)}$ given by the difference of the centroid and $\mathbf{v_C}^{(1)}$.

Figure 5 shows the SGS wavelet decomposition lifting steps that take a mesh from tier 1 to tier 0. In general, the lifting steps are given by:

$$\mathbf{v}^{(\mathbf{k})} = \frac{1}{\alpha}\mathbf{v}^{(\mathbf{k+1})} - \frac{1-\alpha}{m\alpha} \sum_{\forall \mathbf{v} \in \mathcal{E}(\mathbf{v}^{(\mathbf{k+1})})} \mathbf{v} \tag{3}$$

$$\delta^{(\mathbf{k})} = \mathbf{v}_{\mathbf{C}}^{(\mathbf{k+1})} - \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{v}_{\mathbf{i}}^{(\mathbf{k})} \tag{4}$$

with $\mathcal{E}(\mathbf{v}^{(\mathbf{k+1})})$ returning the edge of neighbors of a tier $k+1$ vertex with respect to the connectivity of tier $k+1$. The parameter α of the vertex smoothing operation controls the shape and stability of the wavelet basis functions.

The reconstruction steps are obtained by inverting the decomposition steps and reversing their order. Note that except for the appearance of the wavelet coefficient $\delta^{(\mathbf{k})}$, these steps are the same as the SGS subdivision rules presented in the previous section.

$$\mathbf{v}_{\mathbf{C}}^{(\mathbf{k+1})} = \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{v}_{\mathbf{i}}^{(\mathbf{k})} + \delta^{(\mathbf{k})} \tag{5}$$

$$\mathbf{v}^{(\mathbf{k+1})} = \alpha \mathbf{v}^{(\mathbf{k})} + \frac{1-\alpha}{m} \sum_{\forall \mathbf{v} \in \mathcal{E}(\mathbf{v}^{(\mathbf{k+1})})} \mathbf{v} \tag{6}$$

where $\mathbf{v}_{\mathbf{0}}^{(\mathbf{k})}..\mathbf{v}_{\mathbf{n-1}}^{(\mathbf{k})}$ are the vertices of a tier k cell C and $\delta^{(\mathbf{k})}$ is a wavelet coefficient which is vector-valued like the vertices of the subdivision. $\mathcal{E}(\mathbf{v}^{(\mathbf{k+1})})$ is the set of edge neighbors of $\mathbf{v}^{(\mathbf{k})}$ in the tier $k+1$ mesh. The smoothing rule for tier k vertices (6) remains unchanged.

4.2 Generalized Linear B-Spline Wavelets

Lifted linear B-spline wavelets can be generalized to the slow growing refinement rules. The linear B-spline transform computes the wavelet coefficient first, by predicting cell vertices by centroids, and encoding the difference between the centroids and the actual cell vertex positions as wavelet coefficients. In contrast to the SGS smoothing rules, the original vertices are smoothed only by the wavelet coefficients.

Figure 6 depicts linear B-spline wavelet decomposition from tier 1 to tier 0 which generates one wavelet coefficient, removes its associated vertex and results in a single tier 0 cell.

$$\delta^{(\mathbf{k})} = \mathbf{v}_{\mathbf{C}}^{(\mathbf{k+1})} - \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{v}_{\mathbf{i}}^{(\mathbf{k})} \tag{7}$$

$$\mathbf{v}^{(\mathbf{k})} = \mathbf{v}^{(\mathbf{k+1})} - \frac{1}{m} \sum_{\forall \delta \in \mathcal{W}(\mathbf{v}^{(\mathbf{k+1})})} \delta \tag{8}$$

where n is the number of vertices of cell C, $\mathcal{W}(\mathbf{v}^{(\mathbf{k+1})})$ is the set of newly computed wavelet coefficients derived from

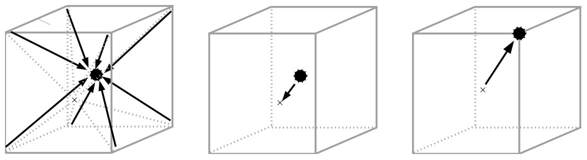

Figure 6. Linear B-spline decomposition of six tier 1 cells with vertices at the corners denoted $\mathbf{v_0}^{(1)}..\mathbf{v_7}^{(1)}$ and a center vertex $\mathbf{v_C}^{(1)}$. (a) Computation of centroid of vertices $\mathbf{v_i}^{(1)}$. (b) A wavelet coefficient $\delta^{(0)}$ given by the difference of the centroid and $\mathbf{v_C}^{(1)}$. (c) Smoothing of $\mathbf{v_i}^{(1)}$ vertices by wavelet coefficients of former edge neighbors.

cell vertices $\mathbf{v}_{\mathbf{C}}^{(\mathbf{k+1})}$ edge connected to $\mathbf{v}^{(\mathbf{k+1})}$ and m is the number of such vertices. The reconstruction is computed as follows:

$$\mathbf{v}^{(\mathbf{k+1})} = \mathbf{v}^{(\mathbf{k})} + \frac{1}{m} \sum_{\forall \delta \in \mathcal{W}(\mathbf{v}^{(\mathbf{k+1})})} \delta \tag{9}$$

$$\mathbf{v}_{\mathbf{C}}^{(\mathbf{k+1})} = \delta^{(\mathbf{k})} + \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{v}_{\mathbf{i}}^{(\mathbf{k})} \tag{10}$$

5 Progressive Compression

A progressive bit stream is constructed from a hierarchy of subdivision wavelets. The resulting stream is optimized for coarse to fine reconstruction operations. Said and Perlman [13] describe an image coder based on set partitioning in hierarchical trees (SPIHT) that is generalized in this paper to the slow growing subdivision refinement structure. The wavelet coefficients in this paper are vector quantities. Each component is quantized to an integer of a certain number of bits before being passed to the SPIHT coder.

5.1 SPIHT Coder

The SPIHT algorithm, like the zerotree algorithm [14], exploits the decay behavior of wavelet coefficients as one proceeds from coarse to fine spatial scales by operating on trees of wavelet coefficients. In general, a parent-child relationship is desired such that the parent coefficient predicts the behavior of its descendants. There is no theoretical framework that can be used to define such a relationship for subdivision wavelets, but intuitively a definition based on the spatial relationships of parent and child analogous to the trees used in the image coding literature can be applied.

The SPIHT algorithm encodes each bit plane separately, from most significant bit to least significant. A predefined traversal order is defined for both the encoder and decoder, and trees of zeros are encoded efficiently by encoding a single bit for the root coefficient. In this way, compression is achieved if the coarse wavelet coefficients are indicative of the behavior of their descendants. Sign bits are encoded into a separate stream.

5.2 Wavelet Coefficient Trees

The parent child relationships are defined with respect to the slow growing refinement structure. These relationships are presented with respect to the tier 0 mesh, since each tier (1 through 3) inserts vertices that can be associated with elements of decreasing dimension in the tier 0 mesh (cells, faces, and edges). All coefficient trees are rooted in mesh elements of the base mesh.

Tier 2 vertices are associated with edge centers of the tier 0 mesh. A coarse wavelet coefficient at tier 3 has only tier 3 descendants that are all associated with that edge.

Tier 1 vertices are associated with face centers of the tier 0 mesh. A tier 2 wavelet at level ℓ has two sets of child coefficients at level $\ell+1$. First, the tier 0 face on which the coefficient lives will be divided into a set of faces, producing one child wavelet for each face. Second, in creating that set of faces, edges will be introduced into the tier 0 face, and each edge will produce one child wavelet.

Tier 0 wavelets are associated with tier 0 cell centers. There are three sets of children for a tier 0 wavelet at level ℓ. First, one wavelet is produced for each subcell that the tier c cell is divided into at tier $\ell+1$. Second, the faces inserted to produce those subcells each produce one wavelet coefficient. Finally, one wavelet is introduced for each internal edge required to subdivide the tier 0 cell.

6 Results

Performance results are presented for two data sets, a synthetic data set and a simulation data set. The vertex coordinates, including scalar field values, were normalized before quantization. In each case, the root mean squared error was computed for the vertices of the mesh using the magnitudes of the vectors from each vertex of the original mesh to the corresponding vertex in the decompressed mesh. For progressive reconstructions, a set number of bits were decompressed, then the fine-scale mesh was reconstructed from these bits.

First, performance results are presented for a noisy ball mesh. The mesh was created by starting with a cube and applying the slow growing subdivision algorithm to obtain a nearly spherical volumetric mesh. Twelve SGS refinements were used from level 0 to level 4, creating 4913 vertices. Random perturbations were applied to these vertices. Each coordinate of each vertex was quantized to 32 bits, resulting in 471648 bits before compression. Figure 6 shows rate

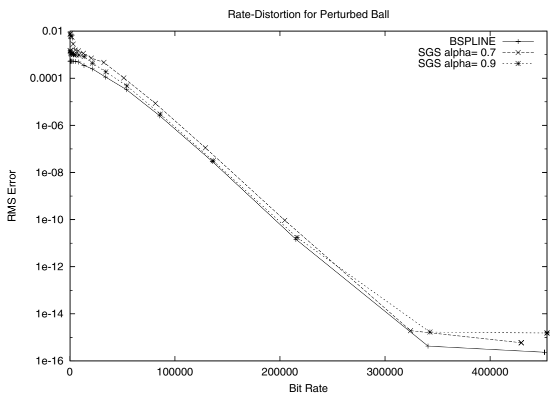

Figure 7. Rate Distortion Curves for the perturbed ball.

distortion curves for both slow growing subdivision and linear B-spline subdivision using the SGS refinement structure for a volumetric ball mesh. SGS compression resulted in 429344 and 454080 bits for $\alpha = 0.7$ and $\alpha = 0.9$ respectively. The linear B-spline transform resulted in 454848 bits. For the slow growing algorithm, α values less than 0.7 resulted in less stable wavelets and decreased compression efficiency. The relatively low compression rates are due to the large perturbations introduced into the mesh, reducing the effectiveness of the SPIHT coder by reducing the correlations between parent and child wavelet coefficients.

Figure 8 shows the external faces of the mesh for three different progressive reconstructions of the data. The superior performance of the B-spline wavelets can be seen at small numbers of bits.

Figure 9 shows the results of applying the wavelet transform to the final cycle of a simulation of laser indirect-drive fusion. The simulation models one quarter of the system and computes the compression of a capsule of fuel inside a hohlraum (a cylindrical container). Laser beams enter the open ends of the hohlraum, striking the inside surface and releasing a uniform bath of radiation that acts on the capsule and causes compression. In the final step shown in figure 9, the internal walls of the hohlraum have been heated and material is ablating from the surface into the cavity inside the hohlraum.

Some domains of the mesh have been eliminated. The opening of the hohlraum is facing the viewer. The capsule is the red spherical area at the rear of the hohlraum. The pseudo-colored field is the material temperature averaged to each node of the subdivision volume. The lower resolution approximations are shown for levels 1 and 2, as well as the full resolution subdivision volume.

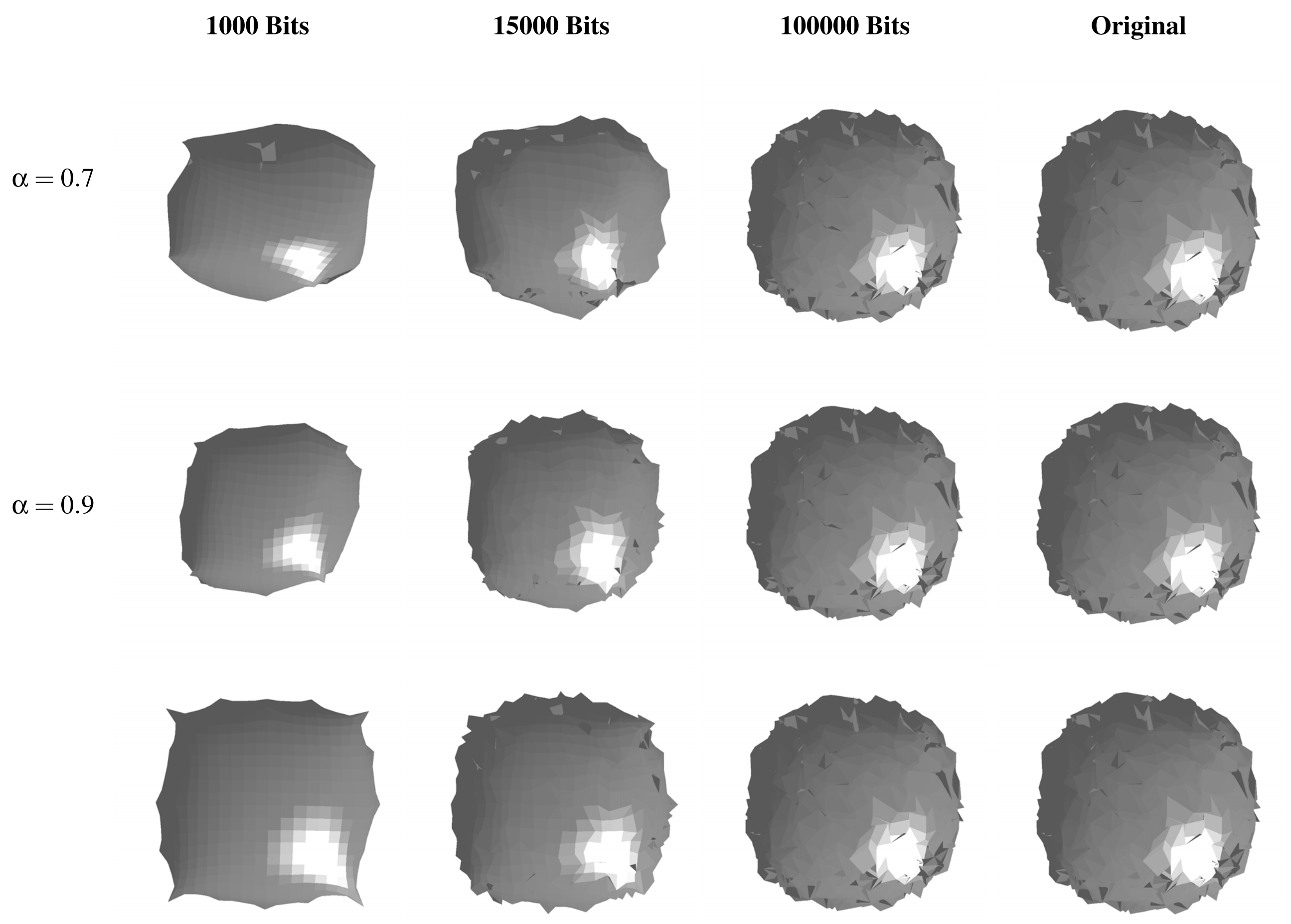

Figure 8. Wavelet reconstructions of a perturbed volumetric (solid) sphere. The top and middle rows show slow growing subdivision with $\alpha = 0.7$ and $\alpha = 0.9$ respectively. The bottom row shows linear B-spline wavelet reconstruction.

Subdivision wavelet transforms require as input a fine scale mesh with subdivision connectivity. Since most finite element meshes are domain decomposed for parallel computation, a remapping must be applied that results in a subdivision volume representing the fields defined on the finite element grid.

The simulation data was decomposed into 32 domains. Each domain contained a regularly connected set of hexahedra with arbitrary vertex locations. A base mesh hexahedron was constructed for each domain, and subdivided 5 times. The fine scale vertex positions were then mapped into the original finite element data and the wavelet transforms were applied. The data set used as input to the wavelet transform contained 134176 vertices.

Figure 6 shows rate-distortion curves for the same data set. The original mesh exhibited great differences in the size of the cells. For example, the hohlraum itself was made up of many layers of thin hexahedral cells, while the empty space between the hohlraum and the capsule was meshed more coarsely. At low numbers of bits, errors in vertex locations caused the mesh to interpenetrate, especially for unstable transforms like the slow growing subdivision wavelets with $\alpha < 0.7$.

7 Conclusion

A complete compression system was implemented on top of the slow growing subdivision algorithm. A slow growing wavelet transform and a generalized B-spline wavelet transform were tested in this system on both synthetic and simulation data. The slow growing wavelets were

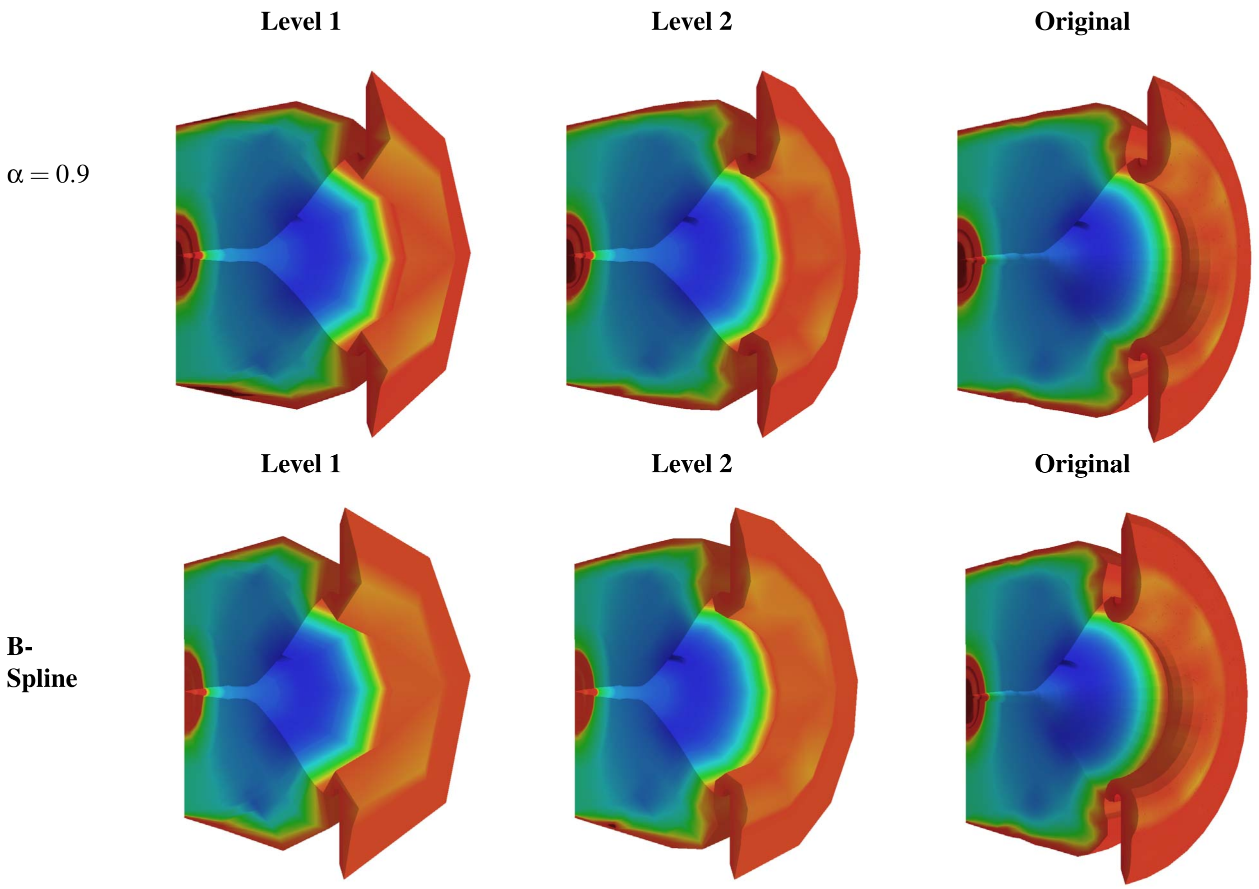

Figure 9. Wavelet reconstructions of an inertial confinement simulation from a view 45° off axis. The top rows show slow growing subdivision with $\alpha = 0.9$. The bottom row shows linear B-spline wavelet reconstruction.

unstable for some settings of the smoothing parameter α. The B-spline transform seemed well behaved.

The system was not tested on base meshes with extraordinary vertices, where special subdivision rules may improve convergence and smoothness. The SGS structure creates vertices with valence other than six in tiers 1 and 2 and the wavelet transform is defined independent of the valence of vertices. However, producing a base mesh from an arbitrary finite element mesh is still an open problem, so the natural domain decomposition of the scientific data was used, resulting in hexahedral base mesh elements. In the case of extraordinary vertices it is probably necessary to solve on the fly for each irregularly connected vertex. Formally proving the stability of subdivision wavelets on volumetric meshes is very difficult due to the large numbers of possible configurations.

The back end compression system does not achieve appreciable gains on the bit streams output by the SPIHT coder. Two areas of improvement are possible. In the system presented in this paper, two bit streams are produced by the SPIHT coder, the sign bits and all other bits. Performance could be improved by further separating the bits into separate streams for the vertex coordinates values and the insignificant coefficients. Secondly, the wavelets themselves can be encoded using a magnitude direction form, under the assumption that the magnitudes will be better predicted by parent wavelet coefficients than the directions.

A progressive representation is not well adapted to local refinement and view-dependent rendering operations. Pascucci *et. al.* [12] have demonstrated a system that provides a globally progressive bit stream, but preserves locality within each level of resolution. A significant research goal is to extend those ideas to the surface and volumetric subdivision settings.

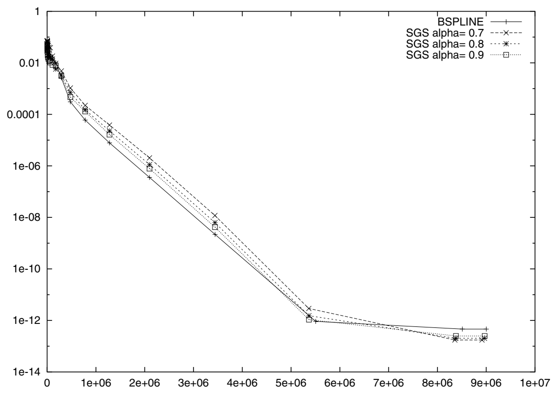

Figure 10. Rate Distortion Curves for Hohlraum and Capsule Simulation.

References

[1] M. Bertram, M. A. Duchaineau, B. Hamann, and K. I. Joy. Bicubic subdivision-surface wavelets for large-scale isosurface representation and visualization. In *Proc. of the 11th Ann. IEEE Visualization Conference (Vis) 2000*, 2000.

[2] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10:350–355, Sept. 1978.

[3] M. H. Gross, L. Lippert, R. Dittrich, and S. Häring. Two methods for wavelet-based volume rendering. *Computers & Graphics*, 21(2):237–252, Mar. 1997. ISSN 0097-8493.

[4] R. Grosso, T. Ertl, and J. Aschoff. Efficient data structures for volume rendering of wavelet-compressed data. In *Winter School of Computer Graphics 1996*, Feb. 1996. held at University of West Bohemia, Plzen, Czech Republic, 12-16 February 1996.

[5] A. Khodakovsky, P. Schröder, and W. Sweldens. Progressive geometry compression. In K. Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, Annual Conference Series, pages 271–278. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[6] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin. MAPS: Multiresolution adaptive parameterization of surfaces. *Computer Graphics*, 32(Annual Conference Series):95–104, Aug. 1998.

[7] L. Linsen, V. Pascucci, M. A. Duchaineau, B. Hamann, and K. I. Joy. Wavelet-based multiresolution with nth-root-of-2 subdivision. In *Journal on Computing, special edition, Dagstuhl Seminar 02201 ' on Geometric Modelling*. Springer-Verlag, Vienna, 2004.

[8] S. Muraki. Approximation and rendering of volume data using wavelet transforms. In *Proc. IEEE Visualization*, pages 21–28, 1992.

[9] S. Muraki. Volume data and wavelet transforms. 13(4):50–56, July 1993.

[10] S. Muraki. Multiscale volume representation by a dog wavelet. In *IEEE Trans. on Visualization and Computer Graphics*, volume 1, pages 109–116, 1995.

[11] V. Pascucci. Slow growing subdivision (SGS) in any dimension: Towards removing the curse of dimensionality. In *Eurographics*, pages 451–460, 2002.

[12] V. Pasuccci and R. J. Frank. Global static indexing for real-time exploration of very large regular grids. In *Proceedings of 14th Annual Supercomputing Conference*, 2001.

[13] A. Said and W. Perlman. A new, fast, and efficient image codec based on set partioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3):243–250, 1996.

[14] J. M. Shapiro. An embedded hierarchical image coder using zerotrees of wavelet coefficients. In J. A. Storer and M. Cohn, editors, *Proceedings DCC'93 (IEEE Data Compression Conference)*, pages 214–233, Snowbird, UT, USA, Apr. 1993.

[15] W. Sweldens. The lifting scheme: A new philosophy in biorthogonal wavelet constructions. In A. F. Laine, M. A. Unser, and M. V. Wickerhauser, editors, *Wavelet applications in signal and image processing III*, volume 2569 of *Proceedings of SPIE*, pages 68–79, 1995.

[16] H. Tao and R. J. Moorhead. Progressive transmission of scientific data using biorthogonal wavelet transform. In R. D. Bergeron and A. E. Kaufman, editors, *Proceedings of the Conference on Visualization*, pages 93–99, Los Alamitos, CA, USA, Oct. 1994. IEEE Computer Society Press.

[17] S. Valette and R. Prost. Wavelet based multiresolution analysis of irregular surface meshes. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):113–122, 2004.

[18] R. Westermann. Compression domain rendering of time-resolved volume data. In *Proc. IEEE Visualization*, 1995.

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.